Simple class with member fields and a default constructor.

Students

Instance field values are stored in the instances.
See StudentApp2.java

Introducing class fields

Students

Class field values are stored only once, in the class itself.

CLASS "MAIN" (StudentApp2)
"Fake" class to process

CLASS STUDENT
A Template or Recipe

Member Fields
lastName
firstName
studentID

Default Constructor
Student()

Class fields
new Student()
new Student()

Student1
Black, Angela
11/11/18

Student2
Garnet, Zachary
2/2/22
19-1

Construct, create, give birth in a default manner.
See StudentApp3.java Introducing member methods.

Students
Class field values are stored only once, in the class itself.

CLASS "MAIN" (StudentApp3)
"Fake" class to process

constuct, create, give birth in a default manner

Student1
Black, Skully
11/11/18

Student2
Garner, Zachary
2/2/22
Introducing "this" keyword inside a member method.

See StudentApp4.java

Class Student

Member fields: lastName, ...

Class field: NO(Student = 2)

Default Constructor: Student()

Member Method: toString()

Self-aware this=instance on which it was called

"Main" class: StudentApp4

Student1
(Student1)
(Black, Quentin, 18)

Student2
(Student2)
(Green, ...)

student1.toString() "Black, Quentin ..."
See StudentApp5.java

Defining our own constructor(s) overrides the default constructor which becomes unavailable.

Class Student

Member fields: lastName, firstName, id
Class fields: noOfStudents = 2
Default Constructor: Student()

Member Method: toString()
Self-aware this instance on which it was called

Constructors:
   Student()
   Student(FirstName, LastName)
   Student(FirstName, LastName, ID, Level)

No more available
Default constructor is no longer available
Do normal "birth" possible
Default constructor overridden

Class StudentApp5

New instance blank to start

New instance: "Jack"
has fields: firstName, lastName with value

ID: 333 333
level: 2
New private static field 9 get and set static methods. The set method is private.

3) Constructors can access private static fields or methods as static.

```
StudentApp7.java
```
4) Methods with the same signature are blocked from inheritance by a lower subclass, e.g., testing().

5) Constructors are not inherited.

Class Rectangle
- Constructors: getLength(), getHeight(),...
- Other methods: constructor(),...
- Member fields: length, height,...

Class Circle
- Constructors: constructor(),...
- Other methods: testing(),...
- Member fields: radius, center,...

3) Access to private member fields can be given if public setter methods are available.

2) Inheritance fields are not inherited.

Introducing
- Rectangle.java
- Circle.java
- GeometricObject.java

Inheriting
- Circle, Rectangle

*Note: testing() is overridden in...

GeometricObject
- Member methods: getCenter(), getSize(),...
- Constructor: constructor()